

We claim:

1. A dynamically loadable stub module, associated with a dynamically loadable kernel module (DLKM), comprising:
 - a base stub module;
 - 5 means for defining DLKM data structures and wrapper functions;
 - means for defining load and unload routines;
 - means for defining metadata structures;
 - means for allowing dynamic loading by DLKM infrastructures; and
 - means for generating a dynamically loadable stub module object file.
- 10 2. The dynamically loadable stub module of claim 1, wherein the means for defining the DLKM data structures and wrapper functions comprises:
 - struct mod_type_data;
 - struct modlink;
 - struct modwrapper; and
 - 15 struct mod_operations.
3. The dynamically loadable stub module of claim 1, wherein the means for defining the DLKM data structures and wrapper functions comprises an autoload statement.
4. The dynamically loadable stub module of claim 3, wherein the autoload statement comprises statements class, and one of stub funcname retfunc, ustub funcname retfunc
20 argnword, and wstub funcname retfunc.
5. The dynamically loadable stub module of claim 1, wherein the means for defining load and unload routines comprises:
 - <module_name>_stub_load (); and
 - <module_name>_stub_unload ().
- 25 6. The dynamically loadable stub module of claim 1, wherein the means for defining metadata structures comprises module version, type, definition, states, and loadtime.
7. The dynamically loadable stub module of claim 1, wherein the means for defining metadata structures comprises a developer-supplied modmeta file.
8. The dynamically loadable stub module of claim 7, wherein the modmeta file is
30 compiled by a modmeta compiler to produce a stub_modmeta.c file.
9. The dynamically loadable stub module of claim 7, wherein metadata is supplied from the associated DLKM.
10. The dynamically loadable stub module of claim 1, wherein the dynamically loadable stub module is included in a kernel data space.

11. The dynamically loadable stub module of claim 1, wherein the dynamically loadable stub module is capable of being statically linked to a kernel executable.
12. The dynamically loadable stub module of claim 1, wherein the data structures comprise:
- 5 struct mod_stub_modinfo; and
 struct mod_stubinfo.
13. The dynamically loadable stub module of claim 12, further comprising the stub routines that use the data structures to manipulate stack frames to transfer control from the dynamically loadable stub module to the associated DLKM.
- 10 14. The dynamically loadable stub module of claim 1, wherein the means for allowing dynamic loading by DLKM infrastructures comprises an ELF section.
15. The dynamically loadable stub module of claim 1, wherein the associated DLKM is a miscellaneous module.
16. A method for dynamic loading of a stub module, comprising:
- 15 modifying a base stub module for an associated DLKM, comprising:
 defining DLKM data structures and wrapper functions for the stub module;
 defining load and unload routines for the stub module;
 defining metadata structures for the stub module;
 providing for dynamic loading of the stub module by DLKM
20 infrastructures; and
 generating a dynamically loadable stub module object file.
17. The method of claim 16, wherein defining DLKM data structures and wrapper functions comprises defining:
- 25 struct mod_type_data;
 struct modlink;
 struct modwrapper; and
 struct mod_operations.
18. The method of claim 16, wherein defining DLKM data structures and wrapper functions comprises providing an autoloader statement comprising class, and one of stub
30 funcname retfunc, ustub funcname retfunc argnword, and wstub funcname retfunc.
19. The method of claim 16, wherein defining metadata structures comprises providing a developer-supplied modmeta file.
20. The dynamically loadable stub module of claim 19, further comprising compiling the modmeta file to produce a stub_modmeta.c file.

21. A computer-readable medium having computer code to implement autoloading stub modules, the code when executed comprising the following steps:
- defining DLKM data structures and wrapper functions for the stub module;
 - defining load and unload routines for the stub module;
 - 5 defining metadata structures for the stub module;
 - providing for dynamic loading of the stub module by DLKM infrastructures; and
 - generating a dynamically loadable stub module object file.